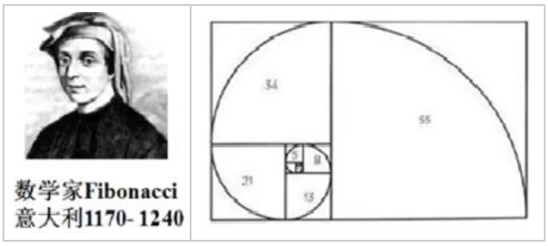
递推方程

2019年1月15日 23:57

一. 递推方程recurrence equation

1. 设序列 $a_0, a_1, \ldots, a_n, \ldots$，简记为$\{a_n\}$，
   一个把 a_n 与某些个 a_i（$i<n$）联系起来的等式
   叫做关于序列 $\{a_n\}$ 的递推方程
   当给定递推方程和适当的初值就唯一确定了序列.
2. Fibnacci数列
   1) 

   2) 1, 1, 2, 3, 5, 8, 13,…
      $f_n = f_{n-1} + f_{n-2}$
      $f_0 = 1,\ f_1 = 1$
   3) $f_n = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^{n+1} - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^{n+1}$
3. Fibnacci数列性质
   1) Σ[i=1:n]fn=f(n+2)-1
   2) Σ[i=1:n]fn ^ 2=f(n+1)f(n)
   3) Σ[i=1:n]f(2i-1)=f(2n)
   4) Σ[i=1:n]f(2i)=f(2n+1)-1
   5) fn=fm*f(n-m+1)+f(m-1)f(n-m) iff n>=m
   6) f(n-1)*f(n+1)=f(n)^2+(-1)^n
   7) gcd(fn,fm)=f(gcd(n,m))
   8) m%n==0 => fm%fn==0

二. 常系数线性齐次递推方程（齐次指H移到左边后，右边=0）

$$\begin{cases} H(n) - a_1 H(n-1) - a_2 H(n-2) - \ldots - a_k H(n-k) = 0 \\ H(0) = b_0, H(1) = b_1, H(2) = b_2, \ldots, H(k-1) = b_{k-1} \end{cases}$$

1. 其中$a_1, a_2, \ldots, a_k$为常数，$a_k \neq 0$，
   称为 k 阶常系数线性齐次递推方程,
   $b_0, b_1, \ldots, b_{k-1}$为 k 个初值
2. 特征方程 $x^k - a_1 x^{k-1} - \ldots - a_k = 0$
   特征方程的根称为递推方程的特征根
   1) 例：Fibnacci
   2) 递推方程 $f_n = f_{n-1} + f_{n-2}$
      特征方程 $x^2 - x - 1 = 0$
      特征根 $\frac{1+\sqrt{5}}{2}, \frac{1-\sqrt{5}}{2}$
3. 定理 1 q 是非零复数，则 q^n 是递推方程的解
   $\Leftrightarrow q$ 是它的特征根

1) 证：　q^n 是递推方程的解

$\Leftrightarrow q^n - a_1 q^{n-1} - a_2 q^{n-2} - \ldots - a_k q^{n-k} = 0$

$\Leftrightarrow q^{n-k}(q^k - a_1 q^{k-1} - a_2 q^{k-2} - \ldots - a_k) = 0$

$\Leftrightarrow q^k - a_1 q^{k-1} - a_2 q^{k-2} - \ldots - a_k = 0$

$\Leftrightarrow q$ 是它的特征根

4. **定理 2**　$h_1(n)$和 $h_2(n)$是递推方程的解，c_1, c_2 为任意常数，则 $c_1h_1(n)+c_2h_2(n)$是递推方程的解.

1) 证明　将 $c_1h_1(n)+c_2h_2(n)$代入递推方程左边，化简后等于 0

2) 推论：若 $q_1, q_2, \ldots, q_k$ 是递推方程的特征根，则 $c_1q_1^n+c_2q_2^n+\ldots+c_kq_k^n$ 是递推方程的解，其中 $c_1, c_2, \ldots, c_k$ 是任意常数.

5. 通解

1) 若对递推方程的每个解 $h(n)$都存在一组常数 $c_1', c_2', \ldots, c_k'$ 使得 $h(n)=c_1'q_1^n+c_2'q_2^n+\ldots+c_k'q_k^n$ 成立，则称 $c_1q_1^n+c_2q_2^n+\ldots+c_kq_k^n$ 为通解.

2) **定理 3**　设 $q_1, q_2, \ldots, q_k$ 是递推方程不等的特征根，则 $H(n)=c_1q_1^n+c_2q_2^n+\ldots+c_kq_k^n$ 为通解.

3) 证：　$H(n)$是解.

设 $h(n)$是递推方程的任意解，

$h(0), h(1), \ldots, h(k-1)$由初值 $b_0, b_1, \ldots, b_{k-1}$ 唯一确定.

$$\begin{cases} c_1'+c_2'+\ldots+c_k'=b_0 \\ c_1'q_1+c_2'q_2+\ldots+c_k'q_k=b_1 \\ \ldots \\ c_1'q_1^{k-1}+c_2'q_2^{k-1}+\ldots+c_k'q_k^{k-1}=b_{k-1} \end{cases}$$

系数行列式　$\prod_{1\le i<j\le k}(q_i-q_j)\neq 0$　当 $q_i \neq q_j$ 时

方程组有唯一解

4) 这个行列式是范德蒙德行列式（每行减去上一行的q1倍，使第一列只有第一行不是0，按第一行展开，再提出公因式π(qi-q1)类似地不断降阶）

6. 重根问题

1) 例 5　$H(n)-4H(n-1)+4H(n-2)=0$

$H(0)=0, H(1)=1$

特征方程 $x^2-4x+4=0$

通解 $H(n)=c_12^n+c_22^n=c2^n$

代入方程得：

$c2^n-4c2^{n-1}+4c2^{n-2}=0$

$c-2c+c=0$,　c 无解.

问题：两个解线性相关.

观察：$n2^n$ 是解，且与 2^n 线性无关

2) **定理 4**　若 q 是递推方程的 e 重特征根，则 $q^n, nq^n, \ldots, n^{e-1}q^n$ 是递推方程的线性无关的解

3) **定理 5**　设 $q_1, q_2, \ldots, q_t$ 是递推方程的不相等的特征根，且 q_i 的重数为 e_i，令

$$H_i(n)=(c_{i1}+c_{i2}n+\ldots+c_{ie_i}n^{e_i-1})q_i^n$$

那么通解　$H(n)=\sum_{i=1}^{t}H_i(n)$

三. 常系数线性非齐次递推方程

1. $H(n)-a_1H(n-1)-\ldots-a_kH(n-k)=f(n),\quad n\ge k, a_k\neq 0, f(n)\neq 0.$

定理 6 设 $\overline{H(n)}$ 是对应齐次方程的通解，$H^*(n)$ 是一个特解，则

2. $$H(n)=\overline{H(n)}+H^*(n)$$

是递推方程的通解.

证（1）$H(n)$是解，代入验证.

（2）设 $h(n)$是解，证明 $h(n)$为一个齐次解与特解 $H^*(n)$之和.

1) $$h(n)-a_1h(n-1)-\ldots-a_kh(n-k)=f(n)$$

$$-)H^*(n)-a_1H^*(n-1)-\ldots-a_kH^*(n-k)=f(n)$$

$$[h(n)-H^*(n)]-a_1[h(n-1)-H^*(n-1)]-\ldots$$

$$-a_k[h(n-k)-H^*(n-k)]=0$$

$h(n)-H^*(n)$是齐次解，即 $h(n)$是一个齐次解与 $H^*(n)$之和.

3. 特解求法
   1) $f(n)$为 n 的 t 次多项式，一般 $H^*(n)$也为 n 的 t 次多项式

      例 7 $a_n+5a_{n-1}+6a_{n-2}=3n^2$

      设 $a_n^*=P_1n^2+P_2n+P_3$，代入得

      $P_1n^2+P_2n+P_3+5[P_1(n-1)^2+P_2(n-1)+P_3]+6[P_1(n-2)^2+P_2(n-2)+P_3]=3n^2$

      i. $$\begin{cases}12P_1=3\\-34P_1+12P_2=0\\29P_1-17P_2+12P_3=0\end{cases}$$

      $P_1=\frac{1}{4}$，$P_2=\frac{17}{24}$，$P_3=\frac{115}{288}$，

      $a_n^*=\frac{1}{4}n^2+\frac{17}{24}n+\frac{115}{288}$

      ii. 通解为 $a_n=c_1(-2)^n+c_2(-3)^n+\frac{1}{4}n^2+\frac{17}{24}n+\frac{115}{288}$
   2) 特征根为1时，特解次数要多设一次，特征根有n重根1时，要多设n次

      例 9 $H(n)-H(n-1)=7n$

      i. 设特解 P_1n+P_2 不行，应设 n^2 次项，因为特征根是 1.

      设 $H^*(n)=P_1n^2+P_2n$，代入 解得 $P_1=P_2=7/2$，

      通解为 $H(n)=c\cdot1^n+\frac{7}{2}n(n+1)=c+\frac{7}{2}n(n+1)$
   3) $f(n)$为指数函数 β^n，若β不是特征根，则特解为 $H^*(n)=P\beta^n$

      例 10 通信编码问题

      $a_n=6a_{n-1}+8^{n-1}$，$a_1=7$

      i. $a^*_n=P\,8^{n-1}$，代入得 $P=4$

      通解 $a_n=c\cdot6^n+4\cdot8^{n-1}$

      代入初值得 $a_n=(6^n+8^n)/2$
   4) 若 β 是 e 重特征根，则特解为 $Pn^e\beta^n$

      例 11 $H(n)-5H(n-1)+6H(n-2)=2^n$，

      $H^*(n)=Pn2^n$，

      代入得

      i. $Pn2^n-5P(n-1)2^{n-1}+6P(n-2)2^{n-2}=2^n$

      解得 $P=-2$

      $H^*(n)=-n2^{n+1}$
   5) 例 12 组合形式

      $a_n-2a_{n-1}=n+3^n$

      $a_0=0$

      设特解为 $a_n^*=P_1n+P_2+P_33^n$，代入

      $(P_1n+P_2+P_33^n)-2[P_1(n-1)+P_2+P_33^{n-1}]=n+3^n$

      $-P_1n+(2P_1-P_2)+P_33^{n-1}=n+3^n$

      $P_1=-1, P_2=-2, P_3=3$

      $a_n=c2^n-n-2+3^{n+1}$

      解得 $c=-1$，$a_n=-2^n-n-2+3^{n+1}$

四. 递推方程其他解法

1. 换元法：尝试换出常系数线性递推方程

例 1 $\begin{cases} a_n^2 = 2a_{n-1}^2 + 1 \\ a_0 = 2 \end{cases} \quad a_n > 0$

   1) 令 $b_n = a_n^2$，代入得

      $b_n = 2\, b_{n-1} + 1$，$b_0 = 4$

      解得 $b_n = 5 \cdot 2^n - 1$，$a_n = \sqrt{5 \cdot 2^n - 1}$

   2) 例：归并排序计算逆序数

      i. 算法：基于二分归并排序的算法

         1. *count*←0; 将L划分成L_1与L_2； /*$L_1[1..n/2]$, $L_2[n/2+1..n]$
         2. 递归处理L_1与L_2；
         3. while *p.next*≠null and *q.next* ≠null do
         4. if *p.key*> *q.key* then /*p,q分别指向L_1与L_2首元素
         5. *count*←*count*+*num*(L_1) /**num*为当前L_1元素数
         6. 移走q指向的元素；q←*q.next*
         7. else 移走p指向的元素；p←*p.next*
         8. if *p.next*=null then 将L_2的剩下的全体元素接在后面
         9. else 将L_1的剩下的全体元素接在后面
         10. 将排好序的全体元素放回到L中

         $T(n)=2T(n/2)+n-1$，$T(n)=O(n\log n)$

      ii. 即先分两波各自排序，再在两波中依次找没排好的。其时间复杂度为

         $T(n) = 2\,T(n/2) + n\text{-}1$， $n = 2^k$

         $T(2) = 1$

      iii. 解 $H(k)= 2\,H(k\text{-}1) + 2^k - 1$

         $H(1) = 1$

         令 $H^*(k) = P_1 k2^k + P_2$，解得 $P_1=P_2=1$,

         $H^*(k) = k2^k + 1$

         通解 $H(k) = C\,2^k + k2^k + 1$,

         代入初值，得 $C = \text{-}1$,

         $H(k) = -\,2^k + k2^k + 1$,

         $T(n) = n\log n - n + 1$

2. 迭代归纳法：把Hn-1用Hn-2迭代，找规律

   1) 适用于非常系数的递推方程

      i. 例 3 计数 $a_1, a_2, \ldots, a_n$ 相乘（可交换）的方法数

         $h(n) = (4n\text{-}6)\, h(n\text{-}1)$

         $h(1) = 1$

$$
\begin{aligned}
h(n) &= (4n-6)\,h(n-1) \\
&= (4n-6)(4n-10)\,h(n-2) \\
&= \ldots \\
&= (4n-6)(4n-10)\ldots 6\cdot 2\cdot h(1) \\
&= 2^{n-1}[(2n-3)(2n-5)\ldots 3\cdot 1] \\
&= 2^{n-1}\frac{(2n-2)!}{(2n-2)(2n-4)\ldots 4\cdot 2} = \frac{(2n-2)!}{(n-1)!}
\end{aligned}
$$

3. 错位排列法，适用于有n-2的

   $D_n = (n\text{-}1)(D_{n-1} + D_{n-2})$

   1) $D_n - nD_{n-1} = -[D_{n-1} - (n-1)D_{n-2}] = \ldots$

4. 差消法，适用于全部历史递推方程

   1) $\begin{cases} T(n) = \dfrac{2}{n}\displaystyle\sum_{i=1}^{n-1} T(i) + n + 1, \quad n \ge 2 \\ T(1) = 0 \end{cases}$

$$nT(n) = 2\sum_{i=1}^{n-1} T(i) + cn^2 \qquad c \text{ 为某个常数}$$

$$(n-1)T(n-1) = 2\sum_{i=1}^{n-2} T(i) + c(n-1)^2$$

2) 将两个方程相减得到

$$nT(n) - (n-1)T(n-1) = 2T(n-1) + O(n)$$

化简得到

$$nT(n) = (n+1)T(n-1) + O(n)$$

变形并迭代得到

$$\frac{T(n)}{n+1} = \frac{T(n-1)}{n} + \frac{c}{n+1} = \cdots = c\left[\frac{1}{n+1} + \frac{1}{n} + \cdots + \frac{1}{3} + \frac{T(1)}{2}\right] = c\left[\frac{1}{n+1} + \frac{1}{n} + \cdots + \frac{1}{3}\right]$$

上面公式中的 c 是某个常数,求和使用了积分作为近似结果,请见图 13.3. 根据积分有

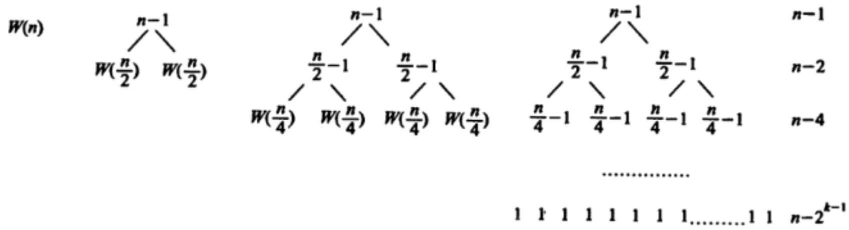
$$\frac{1}{n+1} + \frac{1}{n} + \cdots + \frac{1}{3} \leq \int_2^{n+1} \frac{1}{x}\mathrm{d}x = \ln x \Big|_2^{n+1} = \ln(n+1) - \ln 2 = O(\log n)$$

因此得到原递推方程的解 $T(n) = O(n\log n)$.

3)

算法 快速排序 Quicksort 输入：数组 $A[p..r]$ 输出：排好序的数组 A Quicksort(A,p,r) 1. if $p<r$ 2. then $q \leftarrow$ Partition(A,p,r) 3. $A[p] \leftrightarrow A[q]$ 4. Quicksort($A,p,q-1$) 5. Quicksort($A,q+1,r$)	1. $x \leftarrow A[p]$ //选首元素作为划分标准 x 2. $i \leftarrow p-1$ 3. $j \leftarrow r+1$ 4. while true 5. do repeat $j \leftarrow j-1$ 6. until $A[j] < x$ //$A[j]$是从后向前找到的第一个比 x 小的元素 7. repeat $i \leftarrow i+1$ 8. until $A[i] > x$ //$A[i]$是从前向后找到的第一个比 x 大的元素 9. if $i < j$ // 继续搜索 $A[i]$到 $A[j]$之间的范围 10. then $A[i] \leftrightarrow A[j]$ // $A[i]$与 $A[j]$交换,回到行 4 11. else return j //结束 While 循环

5. 递归树模型方便计算（不含W的作为根，含有W的作为树叶，迭代到初值）
   1) 例：W(n)=2W(n/2)+n-1
   2)



6. 例：分治法的时间复杂度
   1) n 为输入规模，n/b 为子问题输入规模，
   a 为子问题个数，$d(n)$为分解及综合的代价

$$T(n) = aT(n/b) + d(n), \quad n = b^k$$

$$T(1) = 1$$

$$T(n) = a^2T(n/b^2) + ad(n/b) + d(n) = \ldots$$
$$= a^kT(n/b^k) + a^{k-1}d(n/b^{k-1}) + a^{k-2}(n/b^{k-2}) +$$
$$\ldots + ad(n/b) + d(n)$$
$$= a^k + \sum_{i=0}^{k-1} a^i d(n/b^i)$$

$$a^k = a^{\log_b n} = n^{\log_b a}$$

$$T(n) = a^k + \sum_{i=0}^{k-1} a^i d(n/b^i), \quad a^k = n^{\log_b a}$$

   2) (1) $d(n)=c$

$$T(n) = \begin{cases} a^k + c\dfrac{a^k - 1}{a-1} = O(a^k) = O(n^{\log_b a}) & a \neq 1 \\ a^k + kc = O(kc) = O(\log n) & a = 1 \end{cases}$$

二分检索

$$W(n) = W(n/2) + 1$$
$$a = 1, b = 2, d(n) = c$$
$$W(n) = O(\log n)$$

3) (2) $d(n)=cn$

$$T(n)=a^k+\sum_{i=1}^{k-1}a^i\frac{cn}{b^i}=a^k+cn\sum_{i=1}^{k-1}\left(\frac{a}{b}\right)^i$$

$$=\begin{cases} n^{\log_b a}+cn\dfrac{(a/b)^k-1}{a/b-1}=O(n) & a<b \\ n+cnk=O(n\log n) & a=b \\ a^k+cn\dfrac{(a/b)^k-1}{a/b-1}=a^k+c\dfrac{a^k-b^k}{a/b-1}=O(n^{\log_b a}) & a>b \end{cases}$$

归并排序

$W(n) = 2W(n/2) + n-1$

$a = 2, b = 2, d(n) = O(n),$

$W(n)= O(n\log n)$

i.
ii.
iii.
iv.
v.
vi.
vii.
viii. --------我是底线--------